

N91-18976

1990

NASA/ASEE
Summer Faculty Fellowship Program

Marshall Space Flight Center
The University of Alabama

National Launch Strategy
Vehicle Data Management System

Prepared By:	David Cordes
Academic Rank:	Assistant Professor
University:	University of Alabama
Department	Computer Science
NASA/MSFC Division:	Engineering Systems
Branch:	Information SystemsOffice
MSFC Colleague:	Joe Pollock
Contract Number:	NGT-01-002-099 The University of Alabama

National Launch Strategy

Vehicle Data Management System

The NLS/VDMS system was developed as part of the 1990 NASA Summer Faculty Fellowship Program. The system was developed under the guidance of the Engineering Systems Branch of the Information Systems Office, and is intended for use within the Program Development Branch PD34. The development request, and primary system user, was Dr. Jim Steincamp.

The NLS/VDMS system is an on-line database system that permits the tracking of various launch vehicle configurations within the PD office. The system is designed to permit the definition of new launch vehicles, as well as the ability to display and edit existing launch vehicles. Vehicles can be grouped in logical architectures within the system. Reports generated from this package include Vehicle Data Sheets, Architecture Data Sheets, and Vehicle Flight Rate Reports.

System Overview

The system design took the following items into consideration during the course of its development. First, the system must possess a clean user interface. Second, the system should operate on a platform that the user community is familiar with. Third, the system should be straightforward, and contain on-line help for questions that might arise during its execution.

Given these constraints, the following platform was adopted for the development of this package. It will be Macintosh based, as the user community in question is familiar with the Mac, and the Macintosh also provides a clean user interface. Second, it will utilize the ORACLE database package for storing the information within the system. This decision was based on the fact that this database was available on a Mac and was already in the possession of PD34. Third, the system will utilize the SuperCard product for its hypermedia interface. SuperCard provides a hypermedia authoring system for the Macintosh, and is based heavily upon the Macintosh product HyperCard. It is a more flexible product than is HyperCard, and was also already in the possession of the PD34 group.

Initial System Development

The system proposed by the PD 34 group was built in a rapid prototyping environment. Given the time frame involved with the project, a conventional development strategy was not feasible. Instead, initial gathering of

requirements was performed with Dr. Steincamp, and then prototypes of the user interface into the system were generated. Based on Dr. Steincamp's comments regarding these interfaces, the initial system model was generated. Once the model had been established, the individual functions demanded within the model were developed.

Specific system functionality included the following: First, the system must be capable of defining, displaying and editing (with history) individual vehicles within the system. These vehicles are to be composed out of modular components. These components, other system entities, must also be able to be definable, displayable and editable (with history). Furthermore, the vehicles in the system should be able to be grouped into logical sets, called architectures. Finally, the generation of various reports was also fundamental to the project. The system must be able to generate both Launch Vehicle Data Sheets and also Architecture Data Sheets. In addition, the generation of Vehicle Flight Rate Reports was required.

As the design and implementation processes associated with this project are of little concern to the average individual, this report will instead concentrate on introducing some of the tools that were utilized to implement the system. Of particular interest are the SuperCard and ORACLE products. It was the functionality and abilities of these two products that impacted the development of this system the most.

SuperCard Hypermedia Authoring System

SuperCard is a powerful hypermedia authoring system. It is designed as a platform for the development of applications that utilize graphics, text, sound and animation. It is essentially an extension of HyperCard, Apple's commercial hypermedia system. For the NLS/VDMS system, the choice of SuperCard was based on a number of built-in features inherent to the package that permitted the speedy generation of user interfaces. Simple tasks, such as control of the mouse, were trivial within SuperCard. Other systems, such as the X-Windows system, would have demanded much more effort by the programmer.

The SuperCard system is an attempt at an object-oriented system. It is, however, not a true object-oriented system. It does possess some of the basic features inherent to object-oriented systems, such as the communication between objects by events. It lacks any mechanism for inheritance, and does not handle data types in an object-oriented manner. In fact, its data typing abilities are quite limited.

SuperCard did, however, prove invaluable in the prototyping process. The ability to quickly generate sample screens and interaction sequences was ideal for the demands of this project. The built-in primitives provided by SuperCard matched the needs of the development process very well.

The ORACLE Database

The ORACLE Database system is an implementation of a standard relational database model. The system is extremely portable, and is available on a number of hardware platforms. It utilizes the standard SQL query language in its operation. The system is both easy to learn and well-documented.

The ORACLE database was particularly attractive to this project in that it possesses a clean, simple interface into the HyperCard/SuperCard environment. The ORACLE language hypersql can be used as a bridge between SQL and the SuperTalk scripting language. The SQL commands necessary to carry out a specific function in the database can be embedded directly into the scripts of SuperCard.

The database generated for this project was constructed in third normal form. While ORACLE does not demand this of all its applications, third normal form eliminates a number of storage anomalies and also helps to eliminate data redundancy.

The database as it is used in the NLS/VDMS system is a stand-alone product. The ORACLE system does support a distributed database environment for the Macintosh, and the system could be modified so that the product could be run in a networked environment. As this was not part of the original system specifications, it is not currently implemented.

System Evaluation

The completed system is currently running in a Macintosh environment. The complete system utilizes eighteen ORACLE database tables, contains six different windows, 48 cards, 371 card buttons, 521 card fields, 52 card graphics, and 20,432 lines of object scripts. This project was generated in approximately eight weeks, with one week of initial lead time involving learning the Macintosh, ORACLE and SuperCard and one week for generation of the User's and Programmer's manuals.

In looking at the code produced in this project, the final breakdown resulted in an average generation of approximately 64 lines of code per hour. This compares

quite well with the accepted GAO standard of approximately 30 lines of generated code per day.

Additionally, a user's manual and set of developmental documentation has been prepared for the system.

Conclusions

The actual development of this product does not offer up any unique or unexpected conclusions regarding either system prototyping or system development. While it is worth mentioning that the products utilized (i.e. SuperCard) made the development process much easier, these are well-known, standard programming tools. Instead, I choose to concentrate on some concepts that are often ignored by the profession, but have been re-enforced to me over the course of the summer.

1. The ability of our profession to produce a usable product is quite poor. Recent government studies indicate that only eight percent of requested software is ever used. Over twenty-five percent of all government software developments are never even delivered to the users. Quite simply, we are not producing usable, reliable code.

2. The primary sources of error within the software process are in the requirements analysis and system specification phases. The ability to construct a complete, consistent set of functional requirements for a system is very limited. Furthermore, we lack proper techniques for extracting this information from the user population.

3. The overall success or failure of a software product can be attributed to the organizational and management techniques utilized in its development. We lack the ability to effectively organize and manage large-scale software projects.

4. The educational environment in which we develop our programming teams of tomorrow must address these concerns. We cannot simply produce outstanding hackers, but must also be able to output a programming population that possesses a basic set of both organizational and management skills.

This summer has proven very enlightening regarding the actual programming practices that are being used in a real-world situation. This information is going to be carried back directly to my Software Engineering course. By providing the class with a number of real-world scenarios for software development, their appreciation for the organization and management of software projects should be enhanced.